Self-contained Predictive Diagnostic Sensors for Implanter Subsystems

Scott Galica, Katherine Ferrara, Alastair Fisher, William Sweet, David Kirkwood, William MacDonald

Axcelis Technologies

108 Cherry Hill Drive, Beverly Massachusetts, USA 01915

scott.galica@axcelis.com

Abstract

We describe the development of a compact, self-contained sensor which uses machine learning to provide an early warning of failure in implanter subsystems. Using commercially available low power and low-cost MEMS sensors, the system correctly identifies known good and known failing systems with an accuracy of better than 99.9%.

1. Introduction

Micro-electromechanical systems (MEMS) are a broad class of devices encompassing a wide range of functions. Of particular interest are the subcategory of MEMS sensors which include accelerometers, temperature sensors, and magnetometers, among others. The small size of individual MEMS sensor devices allows them to be enclosed in standard electronic component packages. Combining the convenient form factor with their low power requirements makes these sensors easy to embed on compact and inexpensive PCB interfaces. Combining these platforms with embedded micro-controller processing allows for the development of fully integrated sensor platforms at costs and sizes far lower than traditional sensor technologies. This has led to wide adoption of such platforms across a variety of industries.

In parallel to the proliferation of sensor technologies has been the rise and integration of Artificial Intelligence and Machine Learning (AI/ML) processing into sensor platforms. Machine learning is a subfield of artificial intelligence which deals with the development of statistical algorithms for processing data. Sample data is read into an analysis algorithm which in turn generates a secondary algorithm, see figure 1. The secondary algorithm can be used to predict future system behavior based on patterns observed in the training dataset. The adaptive nature of the AI/ML processing techniques make them an invaluable tool for data analysis and reduction.

By coupling these two technologies, we have been able to collect and process high volumes of diagnostic data from our implanter. We have leveraged this data and processing algorithm to serve as an early warning system of failures on end-of-life system and subsystem components.

2. Materials and Methods

As interest in MEMS sensors has grown, so too has the availability of sensors. Several electronic manufacturers and third-party companies now offer integrated development platforms featuring MEMS sensors. We chose to use the SensorTile from ST for our sensor system development. Our choice was driven by three key metrics: flexibility, ease of use, and cost/availability.

The ST SensorTile is a general-purpose evaluation board that comes with eight sensors. Included on the board are a temperature sensor, two 3-axis accelerometers, a 3-axis magnetometer, a barometric pressure sensor, a microphone, a humidity sensor, and a 6 degree of freedom inertial measurement unit (combination accelerometer and gyroscope). All of these components are populated on a single PCB and housed in a plastic enclosure measuring approximately 60 x 35 x 20 mm, see figure 2. The compact size and diversity of included sensors makes the SensorTile well suited to exploratory measurements.

Interfacing with the board can be done over Bluetooth via a smartphone app or via direct programming of the onboard STM32 microcontroller. For our work, the app interface provided sufficient control to collect the required data and allowed us to significantly reduce development time. We used the app to write a custom program for data collection and stored data on the provided internal memory card. The data were transferred to an external computer for processing and AI/ML algorithm development.

SensorTile boards are stocked at several major electronic component distributors as well as through ST directly and at time of writing cost approximately $50 (USD) per unit.

3. Results

3.1 Data Collection

For our proof-of-concept tests we chose to collect and analyze the vibration signatures of one of the implanter wafer handling subsystems. This subsystem was chosen based on relative ease of access to parts on the implanter, number of in-house units available for testing, and the ability to independently verify failure conditions in end-of-life units.

To record our data, we configured a SensorTile to collect high speed acceleration data remotely and store it to the internal memory of the SensorTile board. Data collection was initiated via Bluetooth app and sensors were left in-situ to collect data passively without interrupting regular system operation. Collection duration was primarily limited by the lifetime of the SensorTile battery. As configured for our data collection, this lifetime was approximately 24 hours.

Our initial round of concept data was collected directly on failed components in a benchtop test. We used a SensorTile to collect acceleration data from new and failed subsystem components undergoing replacement. These data, shown in figure 3, show a measurable difference in acceleration magnitude variance between the two unit types. This collection was repeated for different sets of failed components and was shown to be type-consistent across units.

Based on the results of the benchtop tests, we proceeded to on-tool measurements. An in-house subsystem that was identified as being near end of life and beginning to fail was compared to a known-operational subsystem on the same tool. Both subsystems were run through a common set of typical wafer handling tasks and recorded serially using the same SensorTile recording configuration as the benchtop test. Unlike the benchtop test, the sensors were not connected directly to the failing component but rather to the exterior housing of the subsystem. The data are shown in figure 4. The vibration magnitude differences between these datasets, while still measurable, were not as pronounced as those observed in our benchtop test. This was likely due to the weaker vibrational coupling through the exterior case of the subsystem. Despite the reduced signal magnitude, the differences were still sufficient to allow for type classification.

3.2 Algorithm Development

The purpose of the model is to classify which end-of-life subsystems were failing vs still operating properly based on vibration signature without the need for external verification. The measurable differences in acceleration signal between the known-good and known-bad subsystems in both of our test cases provided a robust set of training data for a binary classification algorithm. Even so, the algorithm development was an iterative process.

Our first and simplest model was a decision tree based on acceleration magnitude variance, see figure 5. This model was developed shortly after collecting the benchtop data and before the on-tool data had been collected. Despite yielding good results for the benchtop testing data (figure 5), we found that the model was not flexible enough to accurately predict the on-tool data.

After several iterations of the algorithm, we were able to generate a robust algorithm that performed well across a wider sample of conditions, see figure 6. This algorithm delivered an accuracy of better than 99.9% while minimizing the number of false negatives. The tradeoff for increased accuracy in this case is that the algorithm requires a larger sample size of data to properly classify the subsystem state than our simple decision tree. This requirement is mitigated by using a high sample rate for the accelerometer data collection.

4. Discussion

The final steps in algorithm development are validating the model's accuracy and deploying it on a sensor platform. Testing across several different in-house subsystems has shown robust model performance, but more trials across a wider array of subsystem types are necessary before the model validation is complete.

As we continue to collect validation data for the model, we are also iterating the design of the sensor platform. The SensorTile platform is a very powerful evaluation tool, but it is neither designed nor expected to be a permanent solution. Once the sensor design is settled, the algorithm can be deployed as an embedded program in the sensor firmware. Using a microcontroller based processing or native on-sensor computation (so-called "edge" processing) will allow the sensor to record and report results without the need for external communication. This will allow us to enhance data security while still providing an accurate early warning indicator for system performance.

Conflict of Interest Statement

No funds, grants, or other support was received.

Data Availability Statement

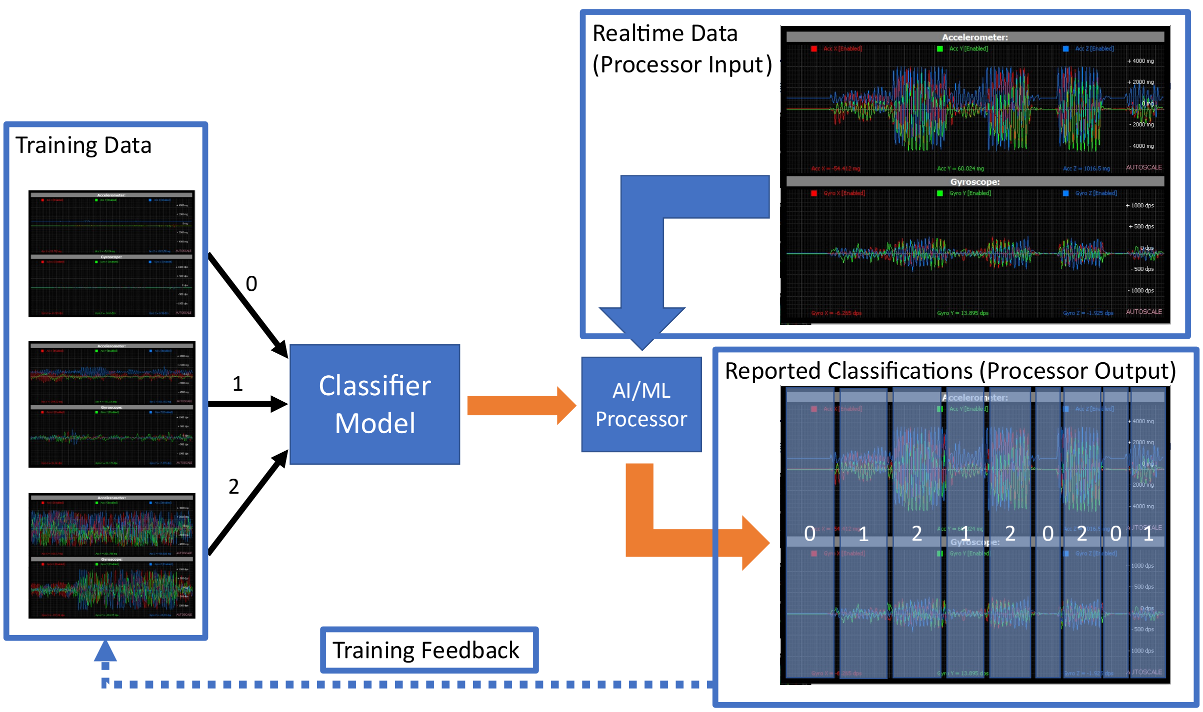
The datasets generated during the study are available from the author upon request.

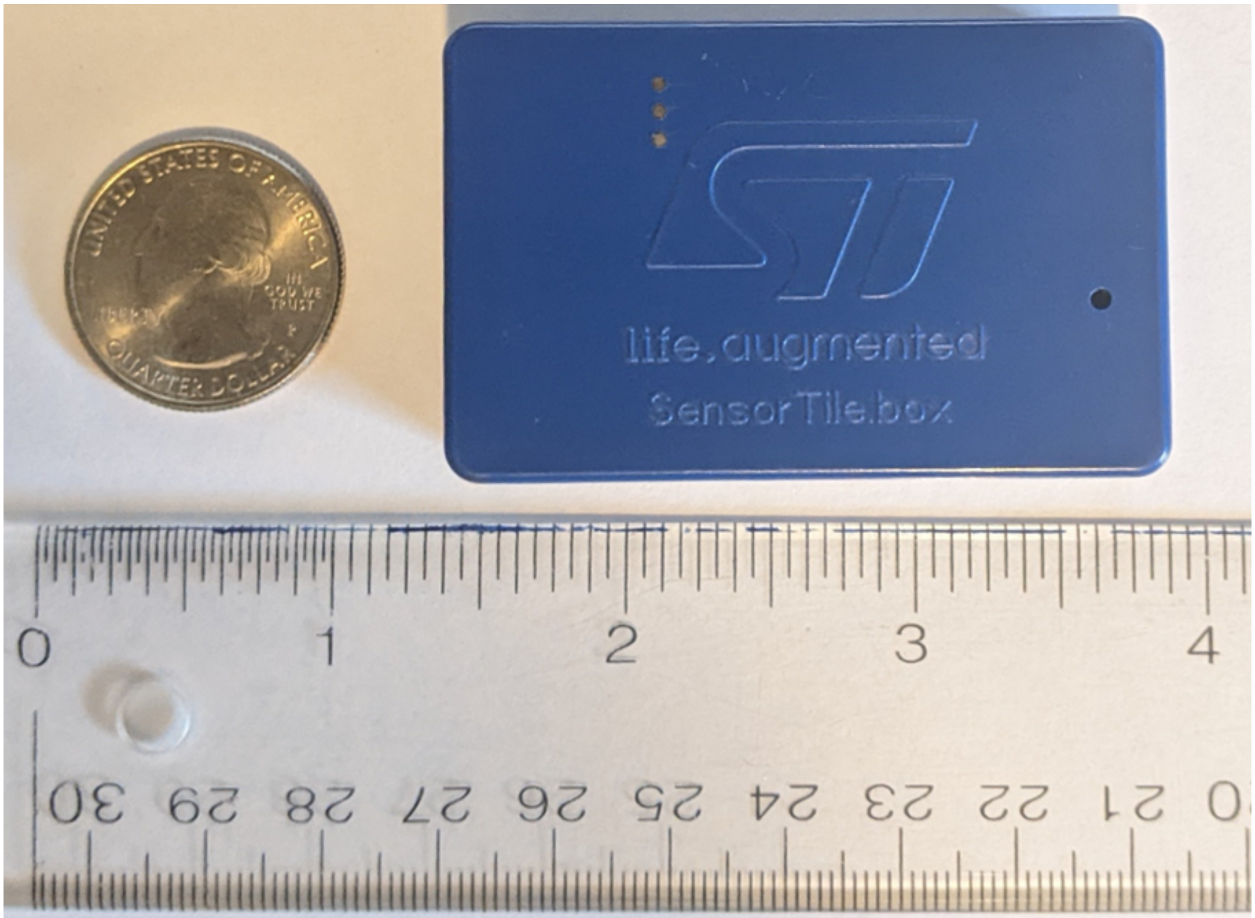
Figure Captions:

1. Schematic of typical AI/ML supervised learning classifier algorithm development process. Training data are collected and fed into a classifier model. The generated model is then deployed on a processor which takes new sample data as inputs and outputs classes assigned to the data by the classifier model. The output classification accuracy can be evaluated and used to further refine the classifier model's algorithm until the desired performance metrics are achieved.

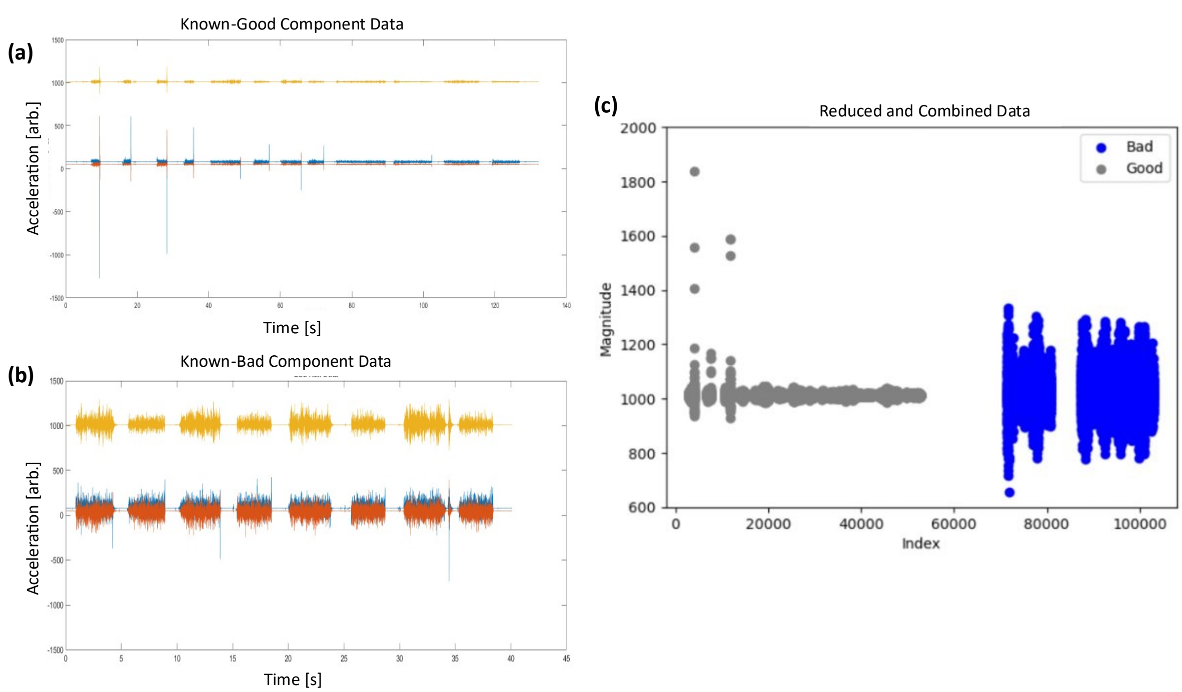
2. ST SensorTile.box MEMS evaluation platform used for data collection. The plastic housing measures approximately 60 x 35 x 20 mm. Ruler and US quarter included for scale.
3. Benchtop accelerometer data collected on isolated subsystem components. (a) Acceleration vector components vs time for a known-good subassembly part. (b) Known-bad subassembly part acceleration vector components vs time. These data are shown on the same vertical axis scaling. (c) Combined and reduced vector magnitude data vs sample index for both known-good and known-bad subassembly parts.
4. Accelerometer data collected from on-tool subassemblies. (a) Acceleration vector components vs time for a known-good subassembly. (b) Known-bad subassembly acceleration vector components vs time. These data are shown on the same vertical axis scaling. (c) Combined and reduced vector magnitude data vs sample index for both known-good and known-bad on-tool subassemblies.
5. Decision tree structure and confusion matrix. (a) Schematic of the decision tree classifier model algorithm. (b) Confusion matrix and key performance metrics for the decision tree classifier model as applied to our test data subset.
6. Confusion matrix and key performance metrics for out iterated classifier model as applied to our on-tool test data.

Figures (in order)

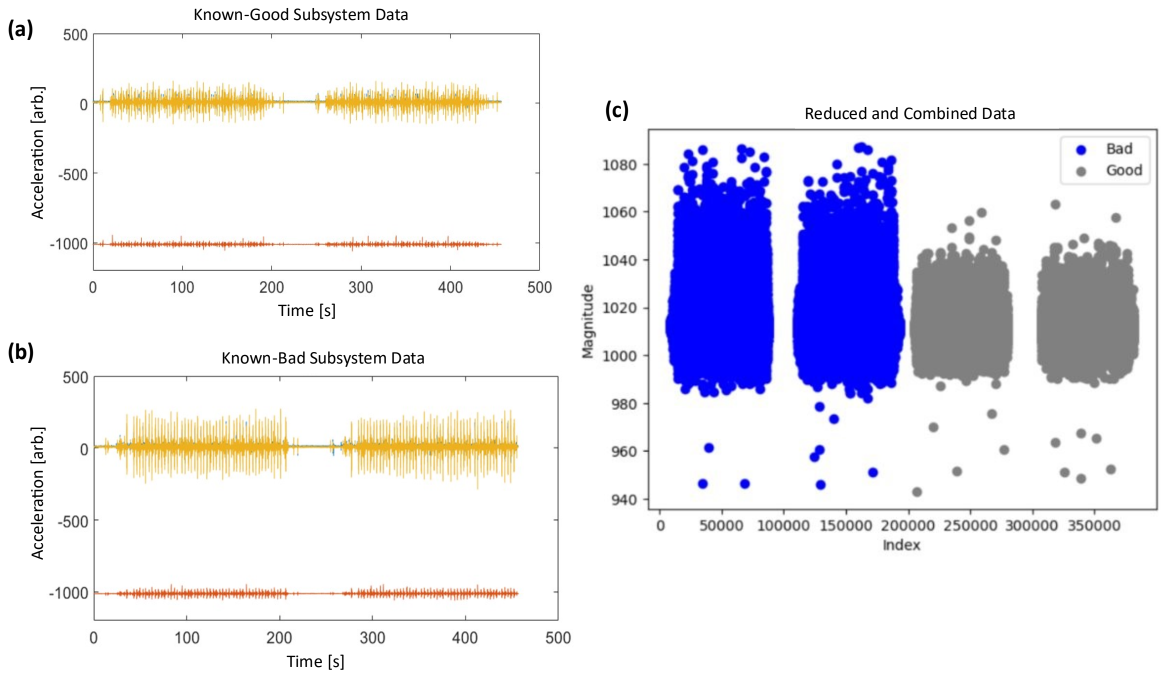




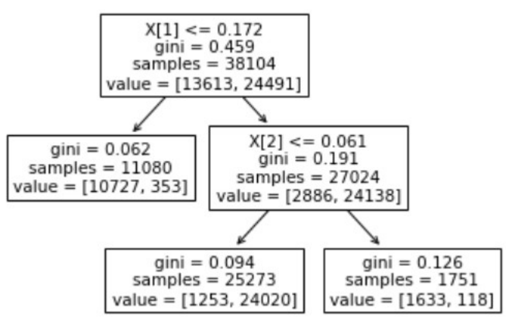
life.augmented
SensorTile.box



(a)
Known-Good Component Data
Acceleration [arb.]
Time [s]
(b)
Known-Bad Component Data
Acceleration [arb.]
Time [s]
(c)
Reduced and Combined Data
Bad
Good
Magnitude
Index



(a)



(b)

Predicted ->	Good	Bad
Good	3,019	298
Bad	123	6,086

- Accuracy: 96%
- Specificity: 98%
- Sensitivity: 91%

Predicted →	Good	Bad
Good	2,178	2
Bad	41	1,164

- Accuracy: 99%
- Sensitivity: 99.9%
- Specificity: 97%